

# PHP 5.4 Performance

Thanks to Dmitry Stogov

Laruence

@laruence

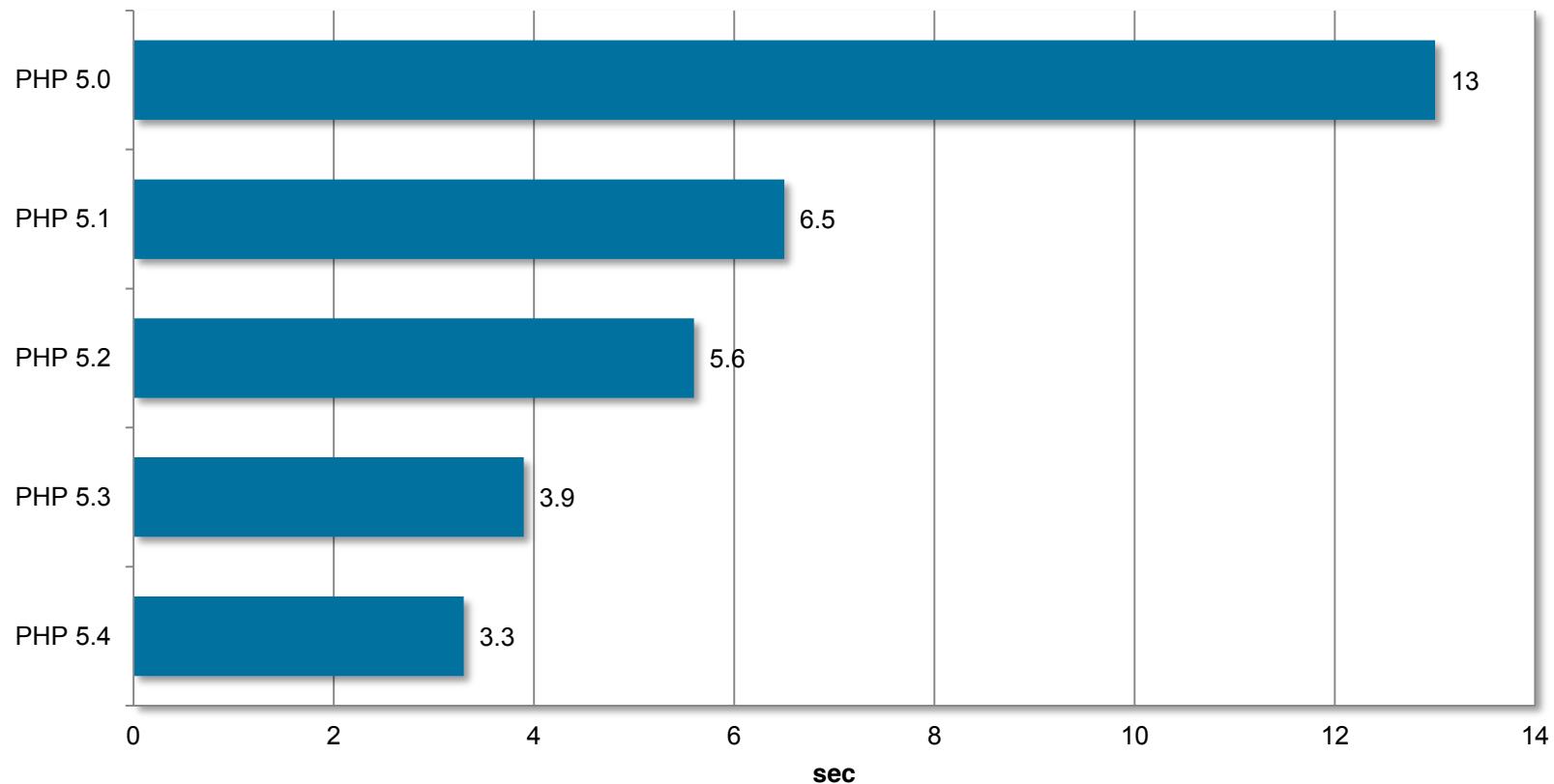
<http://www.laruence.com/>

# PHP

- 1995
- 77.3%
- 5.4
- 4x

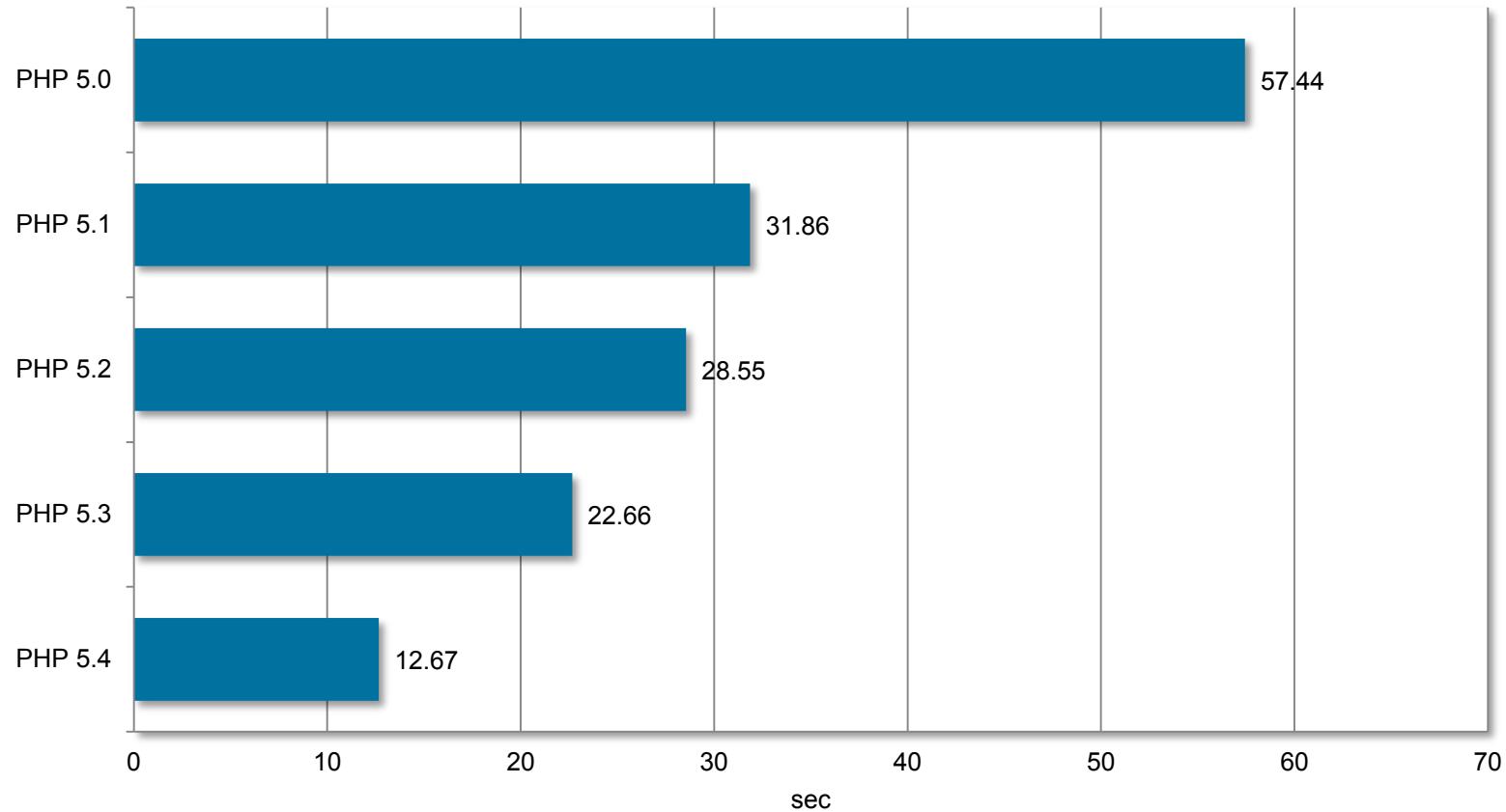
# PHP Performance Evaluation

bench.php



# PHP Performance Evaluation

micro\_bench.php



# New features

- Traits
- Built-in web server
- Array de-referencing
- Class member access on instantiation
- Shortened array syntax
- Binary number format
- ...
- 我们今天只讲5.3到5.4的优化

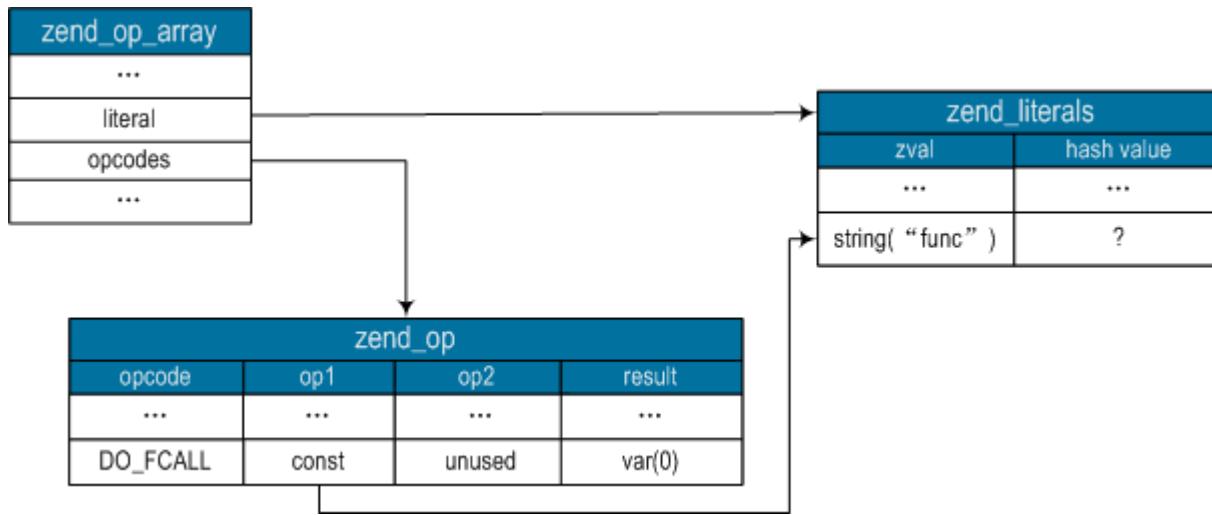
# Delayed Hashtable Allocation

- 很多的HashTable都是空的
- 类的静态属性表, 常量表, 函数表
- 用户定义的空数组
- 推迟对HashTable的Buckets的内存分配, 直到第一个元素被插入

# Literal Tables

- 每一条VM指令, 都包含两个操作数
- 每一个操作数中, 都有一个constant成员, 这在大部分时候, 是没有用的.
- 我们把所有的常量操作数, 都放在一个线性表里面,
- 指令占用的内存数, 从76byte降低到了28byte(X86)

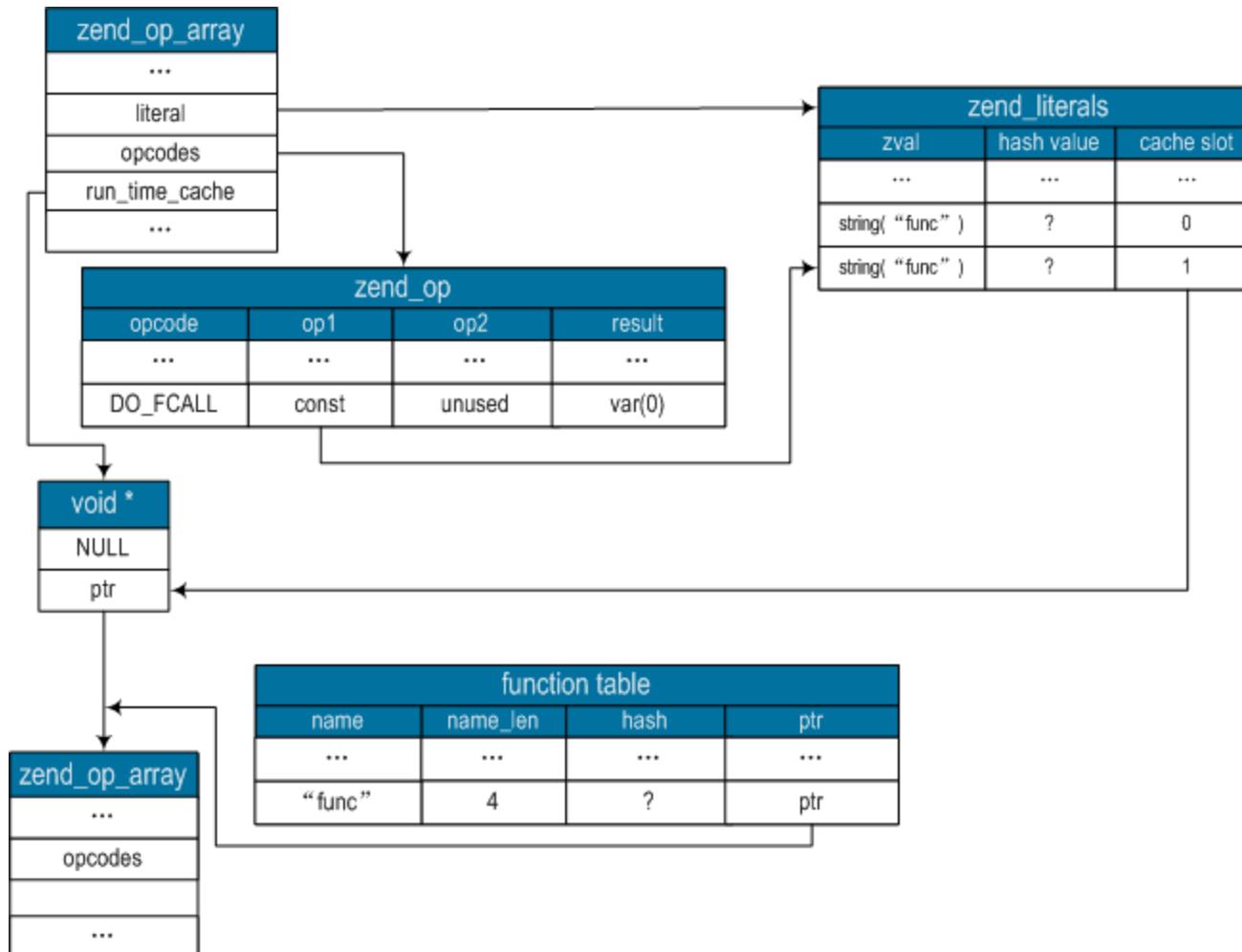
# Literal Tables



# Runtime Binding Cache

- 减少符号查找次数
- 基于 Literal Table.
- 一次查找, 就 Cache 起来
- 多态带来了例外

# Runtime Binding Cache



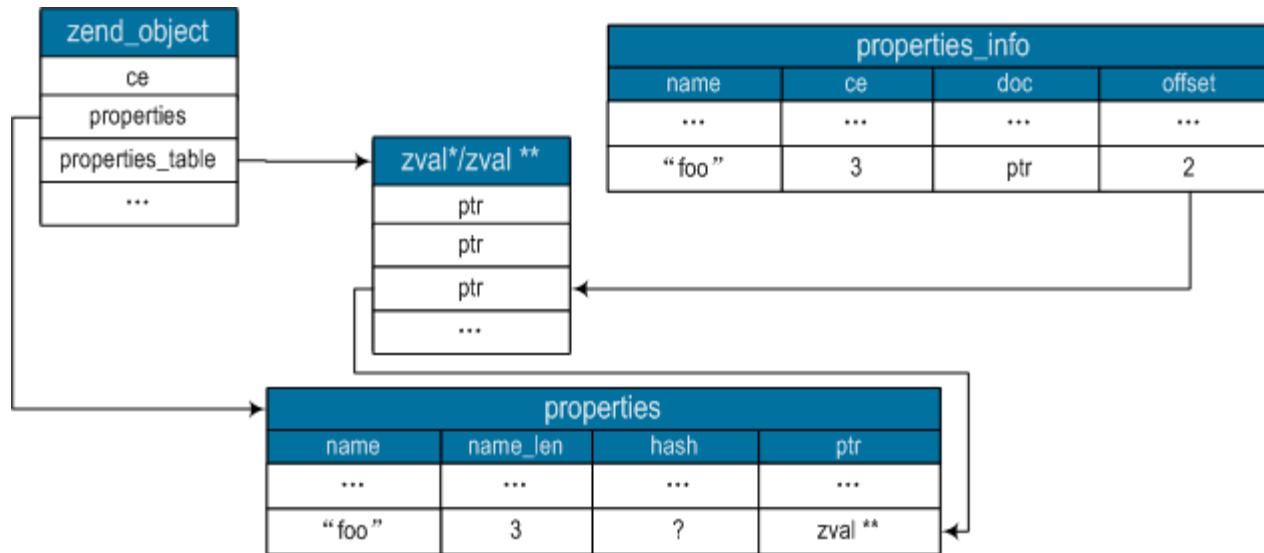
# Internal String

- 编译期已知的字符串
- 对于多个Constant String, 只保存一份
- 使用比较Char \* pointer相等来代替strcmp
- 避免了 strdup 和 free
- 预先计算 Hash Value

# Properties Directly Access

- 查找一个属性, 需要至少2次HashTable 查找
- 我们使用线性表, 来直接保存Properties
- 减少一次HashTable查找

# Properties Directly Access



# Zend Signal

- PHP需要信号屏蔽来保证一些函数的安全
- 信号屏蔽需要系统调用
- HANDLE\_BLOCK/UNBLOCK\_INTERRUPTS需要SAPI自己实现
- 为HashTable的API加入系统调用, 将会带来性能问题

# Request Startup/Shutdown

- 加速扩展的Request启停
- 避免遍历整个扩展列表
- 调用需要调用的

# Ternary

- 三元式每次都会复制要传递的值
- 不支持Copy On Write机制

# FastCGI SAPI Optimization

- FastCGI再接受请求的时候, 因为大量的内存分配导致性能有问题
- 我们改用了Stack Based的内存分配方式
- FastCGI的Parameters Hashtable改用了更加快速的Hash 算法

# More....

- Improved performance of @ (silence) operator
- ZEND\_RETURN -> ZEND\_RETURN, ZEND\_RETURN\_BY\_REF
- ZEND\_RECV result value is always IS\_CV
- ZEND\_CATCH has to be used with constant class name
- Remove break \$var

# Real-life Benchmarks

Apache 2.2			
	No Cache		
Application	5.3.9	5.4.0RC6	gain
Helloworld	11089.8	12297.1	10.89 %
	10678.4	12048.9	12.83 %
	10576.1	11963.7	13.12 %
Drupal	41.9	45.4	8.35 %
	41.1	44.8	9.00 %
	39.9	57.3	43.61 %
Mediawiki	28.6	33.2	16.08 %
	28.4	33.0	16.20 %
	28.2	32.6	15.60 %
Wordpress	44.9	50.4	12.25 %
	44.3	50.5	14.00 %
	43.4	62.0	42.86 %
Joomla	29.6	33.8	14.19 %
	29.4	34.8	18.37 %
	28.6	40.7	42.31 %
Symfony	11.0	11.2	1.82 %
	11.3	11.3	0.00 %
	12.2	13.4	9.84 %

**PHP Performance**  
5.3.9 - 5.4.0RC6

## Hardware & Environment

Dell R710

CPU - Intel Quad core @ 2.26Ghz (x2) L5520

Memory - 12GB RAM

HD - 147GB SAS RAID 1

NIC - 1Gbps Intel

Windows 2008 R2 - SP1

php-web01.ctac.nttest.microsoft.com

# Q&A